

outputting the modular 2^l result.

3. A method for producing a shortened representation of a collection of bits, comprising the steps of:

inputting a collection of "n" bits;
summing a key having at least "n" bits with the collection of bits to produce a sum;
squaring the sum to produce a squared sum;
repeating the previous three steps at least once to produce a plurality of squared sums, where a different key is used each time the steps are repeated;
summing the plurality of squared sums to produce a summation;
performing a modular "p" operation on the summation, where "p" is a first prime number greater than 2^n to produce a modular "p" result;
performing a modular 2^l operation on the modular "p" result to produce a modular 2^l result where, "l" is less than "n"; and
outputting the modular 2^l result.

Remarks

Submission of formal drawings in this case is deferred until such time as this case is allowed.

Claims 1-3 are now in this case. Claims 1-3 have been amended.

Regarding the specification, the title of the specification was described as nondescriptive in the Office action. Applicants respectfully disagree with this conclusion; however, in order to expedite prosecution the title has been changed to Efficient Universal Hashing Method.

The specification was objected with regard to page 1, lines 17-18 and page 2, lines 12-13. The Office action stated that the inequality signs impose no bounds on the probabilities with respect to ϵ . Applications respectfully disagree with the statement. ϵ cannot be greater than 1 because the probabilities which are less than or equal to ϵ cannot exceed 1.

In reference to page 1, line 23, the Office action stated that it was unclear why n is called a domain. The references to “domain” and “range” have been removed from the paragraph on page 1, lines 25-26, and the paragraph on page 4, lines 1-12.

In reference to page 4, line 21, the Office action stated that it was unknown what ϵ stands for. Applicant is unable to locate ϵ at the cited location; however, ϵ was clearly defined in Equations 1, 2 and 3.

Regarding the paragraph beginning on page 6, line 17 and ending at page 7, line 10, a typographic error was corrected by change the SQ_1 to SQ_i .

In reference to page 8, Equation 9, the Office action stated that it is unknown what R stands for. R is defined on page 8, line 12 where R is defined as an abelian group.

In reference to the claims, the Office action stated that it is unknown if p stands for a prime number. The claims have been amended to state that p is a prime number. Support for this amendment may be found in the Specification on page 4, lines 9-10.

Claim 1 was rejected under 35 USC § 103(a) as being unpatentable over Jueneman (ref. 4). Applicants respectfully traverse this rejection. Claim 1 requires that p is **greater than 2^n** . Reference Jueneman indicates that p is **less than $2^m - 1$** . This does not disclose or suggest p being **greater than 2^n** as required by claim 1 as well as claims 2 and 3. Therefore, it is respectfully submitted that independent claim 1 is patentable over Jueneman under 35 USC § 103(a).

Claim 2 was rejected under 35 USC § 103(a) as being unpatentable over Takaragi et al. (U.S. Patent 6,122,375). Applicants respectfully traverse this rejection. As discussed above with regard to claim 1, the references, whether taken alone or in combination, do not disclose or suggest that p is **greater than 2^n** . Additionally, the values X_1 and Y_1 of Takaragi et al. **are not keys** but are the upper 32 bits of **data** and lower 32 bits of **data**, respectively from a 64 bit frame E_q (Takaragi et al., column 17, lines 32-35 and lines 45-50). Claim 2 requires that a **first key** and a **second key** are used in the hashing operation. Since Takaragi et al. **discloses using data rather than keys**, Takaragi et al. does not suggest or disclose using two keys as required in the hashing operation of claim 2. Additionally, reference Jueneman does not disclose or suggest using two keys as required in the hashing operation of claim 2. Since none of the references, whether taken alone or in

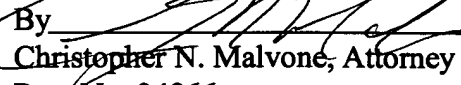
combination, disclose that p is **greater than 2^n** and that a **first key** and a **second key** are used in the hashing operation, it is respectfully submitted that independent claim 2 is patentable over Takaragi et al. under 35 USC § 103(a).

Claim 3 was rejected under 35 USC § 103(a) as being unpatentable over Jueneman's method as taught in the Schneier reference and further in view of Rohatgi et al. (U.S. Patent 5,625,693). Applicants respectfully traverse this rejection. As discussed above with regard to claim 1, the references, whether taken alone or in combination, do not disclose or suggest that p is **greater than 2^n** as required by claim 3. Regarding Rohatgi et al., this reference does not suggest or disclose summing a plurality of **squared sums** as required by claim 3. Rohatgi et al. discloses summing a plurality of **products** (Rohatgi et al., column 10, line 15). Summing a plurality of **products** does not disclose or suggest summing a plurality of **squared sums** as required by claim 3. Since none of the references, whether taken alone or in combination, disclose that p is **greater than 2^n** and that the summation is of a plurality of **squared sums**, it is respectfully submitted that independent claim 3 is patentable over Jueneman's method as taught in the Schneier reference and further in view of Rohatgi et al. under 35 USC § 103(a).

Reconsideration and early allowance of the claims in this case are now requested. If there are any other outstanding issues, the Examiner is invited to contact Applicant's attorney at 973-386-2992.

Respectfully,

Sarvar Patel
Zulfikar Amin Ramzan

By 
Christopher N. Malvone, Attorney
Reg. No. 34866
973-386-2992

Date:

7/12/01

**MARKED UP VERSION OF
SERIAL NO. 09/175178**

Replace the title with the following: **EFFICIENT UNIVERSAL HASHING METHOD.**

Replace the paragraph on page 1, lines 23-26 with the following:

The number of bits contained in the longer unhashed string is " n ", ~~and is called a domain.~~
The number of bits in the shorter or hashed string is " l ", ~~and is often referred to as the range of the hashing function.~~ A hashing function that satisfies Equation (1) is often referred to as ϵ universal.

Replace the paragraph on page 4, lines 1-12 with the following:

In one embodiment of the invention, as illustrated by Equation (6), a hashing of a message " m " is performed by summing the message string with a key string " a " and then forming the square of that summation. A modular " p " operation performed on the result of the squaring operation and a modular 2^l operation is performed on the result of the modular " p " operation. In this case, both " m " and " a " are of the same length, that is, " n " bits or " w " words long. It should be noted that " a " may be longer than " n " bits, but " n " bits is preferable. The value " l " refers to the length in bits of the shortened string that results from the hashing, ~~and is referred to as the range.~~ The value " p " is selected as the first prime number greater than 2^n where " n " is the number of bits in the message string " m ". It should be noted that Equation (6) provides a hashing method that satisfies Equations (1) and (2), that is, the hashing method of Equation (6) is Δ universal.

Replace the paragraph beginning at page 6, line 17 and ending at page 7, line 10 with the following:

FIG. 3 illustrates a method for performing the $\epsilon \Delta$ universal hashing method described by Equation (8). In step 170 index " i " is set equal to 1 and the variable SUM is set equal to 0. In step 172 the value of " k " is inputted. " k " is equal to the number of strings or messages that will be inputted to produce a single shortened message. In step 174 message or string m_i is separated, and in step 176 input key a_i is inputted. It should be noted that message or string m_i and input key a_i are of equal length and have " n " bits composing " w " words. Key " a_i " is a random or pseudo-random number and may be longer than " n " bits, but " n " bits is preferable. Preferably, a_i is a random

number. Random numbers can be generated from many sources such as pseudo-random generators. In step 178 sum s_i is formed by forming the sum of message m_i and key a_i . In step 180 the square of s_i is set equal to variable SQ_i . In step 182 the variable SUM is set equal to the variable SUM plus SQ_i . In step 184 the value of "i" is checked to determine if it is equal to the value "k". If it is not equal to the value "k", step 186 is executed where the value of index "i" is incremented by "1" and then step 174 is executed. If in step 184 the value of "i" is determined to be equal to "k", step 188 is executed where a modular "p" operation is performed on the current value of the variable SUM. As discussed previously, the value "p" is the next prime number greater than the value 2^n ; however, "p" may be a larger prime which may degrade performance. In step 190 a modular 2^l operation is performed on the results produced in step 188. Once again, "l" is the number of bits composing the output string or message. In step 192 the shortened message or string of "l" bits is outputted. It should be noted that the process of FIG. 3 reduced "k" messages of "n" bits each to one message of "l" bits. It should also be noted that the hashing method of FIG. 3 is a $\epsilon \Delta$ universal hashing method that satisfies the properties of Equations (1) and (2).

Amended claims 1-3

1. A method for producing a shortened representation of a collection of bits, comprising the steps of:

inputting the collection of "n" bits;
 summing a key having at least "n" bits with the collection of bits to produce a sum;
 squaring the sum to produce a squared sum;
 performing a modular "p" operation on the squared sum, where "p" is at least as large as a first prime number greater than 2^n to produce a modular "p" result;
 performing a modular 2^l operation on the modular "p" result to produce a modular 2^l result where, "l" is less than "n"; and
 outputting the modular 2^l result.

2. A method for producing a shortened representation of a collection of bits, comprising the steps of:

inputting the collection of "n" bits;

summing a first key having at least “n” bits with the collection of bits to produce a first sum;
squaring the first sum to produce a squared sum;
summing the squared sum with a second key having at least “n” bits to produce a second sum;
performing a modular “p” operation on the second sum, where “p” is at least as large as a first prime number greater than 2^n to produce a modular “p” result;
performing a modular 2^l operation on the modular “p” result to produce a modular 2^l result where, “l” is less than “n”; and
outputting the modular 2^l result.

3. A method for producing a shortened representation of a collection of bits, comprising the steps of:

inputting a collection of “n” bits;
summing a key having at least “n” bits with the collection of bits to produce a sum;
squaring the sum to produce a squared sum;
repeating the previous three steps at least once to produce a plurality of squared sums, where a different key is used each time the steps are repeated;
summing the plurality of squared sums to produce a summation;
performing a modular “p” operation on the summation, where “p” is at least as large as a first prime number greater than 2^n to produce a modular “p” result;
performing a modular 2^l operation on the modular “p” result to produce a modular 2^l result where, “l” is less than “n”; and
outputting the modular 2^l result.